

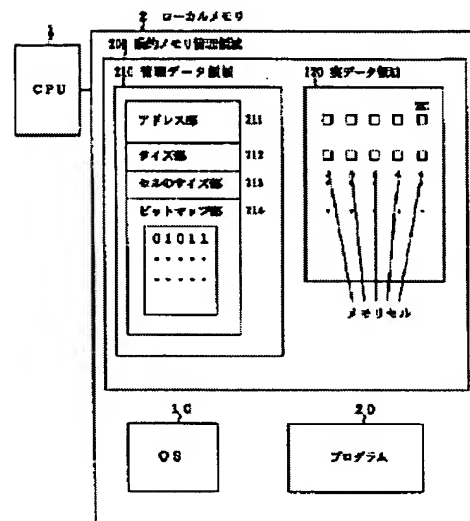
MEMORY MANAGING SYSTEM

Patent number: JP8221317
Publication date: 1996-08-30
Inventor: SHIMOO MANABU
Applicant: TOKYO SHIBAURA ELECTRIC CO
Classification:
- international: G06F12/02; G06F12/02; (IPC1-7): G06F12/02
- european:
Application number: JP19950026716 19950215
Priority number(s): JP19950026716 19950215

Report a data error here

Abstract of JP8221317

PURPOSE: To improve the utilization efficiency of memory and to shorten processing time in the management of variable length memory by dividing a part of local memory into plural memory cells and managing the state of use/unuse state of the memory cell. **CONSTITUTION:** The working state of each memory cell 221 for real data is managed by a bit map part 214 in a managing data area 210. When a 0 bit in the bit map part 214 represents the unuse of a corresponding memory cell 221 for the real data, and a 1 bit represents the aquisition of dynamic memory, an OS 10 checks sequentially each bit in the bit map part 214 in the managing data area 210 from a leading bit, and searches a place where the 0 bit exists. When the 0 bit is found out, a working state is set by turning the bit to 1, and the memory cell 221 for the real date in accordance with the bit is allocated by delivering the address so as to be used by the program 20.



Data supplied from the esp@cenet database - Worldwide

THIS PAGE BLANK (USPTO)

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 8 - 2 2 1 3 1 7

(43) 公開日 平成 8 年 (1996) 8 月 30 日

(51) Int. Cl. ⁶

識別記号

庁内整理番号

F I

技術表示箇所

G 0 6 F 12/02

5 3 0

G 0 6 F 12/02

5 3 0 B

審査請求

未請求

請求項の数 2

O L

(全 6 頁)

(21) 出願番号 特願平 7-26716

(22) 出願日 平成 7 年 (1995) 2 月 15 日

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町 72 番地

(72) 発明者 下尾 学

東京都府中市東芝町 1 番地

株式会社東芝

府中工場内

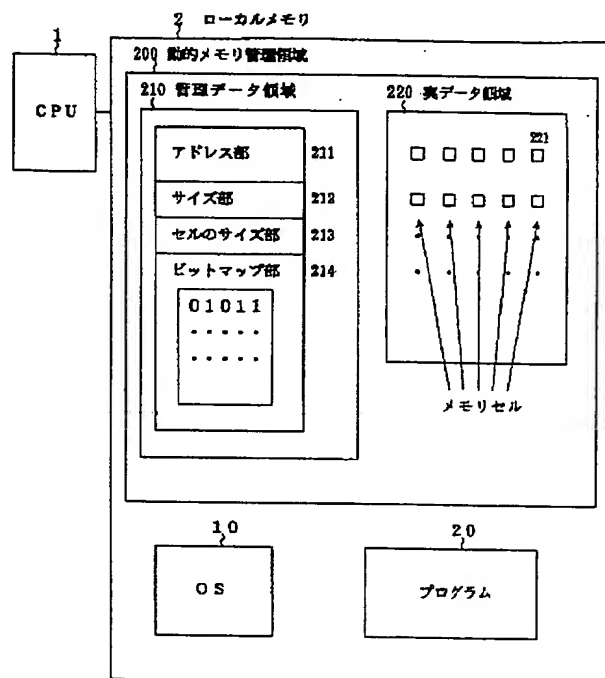
(74) 代理人 弁理士 猪股 祥晃

(54) 【発明の名称】 メモリ管理方式

(57) 【要約】

【目的】 ビットマップを使用して動的メモリ管理を行い、メモリの利用効率の向上及び可変長メモリ管理における処理時間の向上を図ったメモリ管理方式を提供すること。

【構成】 マイクロプロセッサからの指令に応じデータのローカルメモリ内のメモリの割り付けと開放を行なうメモリ管理方式において、ローカルメモリの一部を固定長の複数のメモリセルに分割し、複数のメモリセルにそれぞれビットを対応させるとともに当該メモリセルの使用・未使用の状態を示すビット値として記憶するビットマップを管理データ領域に設け、このビットマップを管理してメモリの割り付けと開放を行っているので、固定長メモリブロックのメモリ管理では、メモリブロックのサイズやその個数に拘らず効率的なメモリ管理が可能となり、また可変長メモリブロックのメモリ管理では、処理が一定の速度で高速に実施することが可能となる。



【特許請求の範囲】

【請求項1】 マイクロプロセッサからの指令に応じデータのローカルメモリ内のメモリの割り付けと開放を行なうメモリ管理方式において、前記ローカルメモリの一部を固定長の複数のメモリセルに分割し、前記複数のメモリセルにそれぞれビットを対応させるとともに当該メモリセルの使用・未使用の状態を示すビット値として記憶するビットマップを管理データ領域に設け、このビットマップを管理してメモリの割り付けと開放を行なうことを特徴とするメモリ管理方式。

【請求項2】 マイクロプロセッサからの指令に応じデータのローカルメモリ内のメモリの割り付けと開放を行なうメモリ管理方式において、前記ローカルメモリの一部を固定長の複数のメモリセルに分割し、前記複数のメモリセルにそれぞれビットを対応させるとともに当該メモリセルの使用・未使用の状態を示すビット値として記憶するビットマップと、前記ビットマップにおける未使用の状態を示すメモリセルの先頭位置およびこの先頭位置のメモリセルに連続する未使用の状態のメモリセルの個数を管理データ領域に設け、メモリの割り付けと開放を行うことを特徴とするメモリ管理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、マイクロプロセッサのオペレーティングシステムでローカルメモリ上のメモリブロックを管理するメモリ管理方式に関するものである。

【0002】

【従来の技術】マイクロプロセッサのオペレーティングシステムでローカルメモリ上の固定長並に変長の従来のメモリ管理方式を図5並に図6を参照して説明する。

(1) ローカルメモリ上にあるメモリ域を固定長メモリブロックに分割し、各メモリブロックの割り付けと開放の管理をする場合、図5に示すように各メモリブロック102(メモリブロック1, 2, …n)をリンク情報111、状態112、データ113で構成する。リンク情報111と状態112を管理データ114と呼ぶ。一方、メモリブロック102中のデータ113を実データ115と呼ぶ。1はマイクロプロセッサ(CPU)、10はオペレーティングシステム(OS)である。

【0003】このリンク情報111と状態112を使って、以下のようにしてメモリブロック102の管理を行う。未使用のメモリブロックは、未使用のポインタ101を先頭とし、各メモリブロック102のリンク情報111を使うリンクを構成する。プログラム20からメモリの獲得要求があった場合は、未使用のリンクからメモリブロック102を獲得し、状態を使用中にした後、その実データ115をプログラム20によって使えるように割り付けする。

【0004】使用中のメモリブロックは、プログラム2

0の中で利用されているため未使用のポインタ101によって管理されることはない。しかし、管理データ114を含むデータそのものは動的メモリ管理領域100に存在する。プログラム20が獲得していたメモリブロック102を開放する場合は、そのメモリブロック102の状態112を未使用に設定し、未使用のポインタ101によって構築されるリンクの先頭または最後につなぐ。

【0005】したがって、従来技術による固定長メモリブロックの管理では、各メモリブロック中の実データ115の大きさが管理データ114の大きさより小さい場合、実際にメモリの割り当てに使用される実データ115よりこれを管理する管理データ114の方が大きくなる可能性がある。

【0006】(2) ローカルメモリ上にあるメモリ域を可変長メモリブロックとして割り付けと開放の管理をする場合、図6に示すように各メモリブロック122(メモリブロック1, 2, …n)をリンク情報131、状態132、サイズ133、データ134で構成する。リンク情報131、状態132、サイズ133の3個を合わせて管理データ135と呼ぶ。一方、メモリブロック122中のデータ134を実データ136と呼ぶ。

【0007】このリンク情報131、状態132、サイズ133を使って、以下のようにしてメモリブロックの管理を行う。可変長メモリブロックの各メモリブロック122(メモリブロック1, 2, …n)は、可変長メモリ管理ポインタ121を先頭として各メモリブロック122のリンク情報131を使うリンクを構成する。この場合、未使用のメモリブロックと使用中のメモリブロックが混在しているため、未使用のメモリブロックを探すために、各メモリブロック122をリンクの先頭から1つ1つたどってそのメモリブロック122の状態132を検査し割り当てが可能か否か判断する。

【0008】プログラム20からメモリの獲得要求があった場合は、可変長メモリ管理ポインタ121のリンクからプログラム20が要求したサイズが収まる大きさの未使用のメモリブロック122を探し、そのメモリブロック122の状態132を使用中にした後、その実データ136をプログラム20によって使えるように割り付けする。このとき、メモリブロック122の実データ136がプログラム20が要求したサイズより大きい場合、2つのメモリブロック122に分割し、一方をプログラム20に割り付けし、もう一方を未使用のメモリブロック122として可変長メモリ管理ポインタ121にリンクする。分割によって2つのメモリブロック122のサイズ133は、それぞれの大きさに変更する。

【0009】プログラム20が獲得していたメモリブロック122を開放する場合は、そのメモリブロック122の状態132を未使用に設定し、リンクの前後のメモリブロックが未使用の状態であれば、1個の大きなメモ

リブロックに組み立てる。

【0010】このようにして可変長メモリブロックを管理するために、リンクの始めの方にあるメモリブロック122と後方にあるメモリブロック122では、後方のメモリブロック122の方が検査に時間がかかり、割り付けの処理時間に大きな違いが出てくる。特に、割り当て可能なメモリブロックより少ない大きさのメモリ獲得要求があったとき、そのメモリブロックを2つに分割し、要求に最適な大きさにするため、メモリブロックの個数が増え後方のメモリブロックの検査に要する時間に拍車をかけることになる。

【0011】

【発明が解決しようとする課題】ところで、上記した従来技術では下記のごとき欠点がある。

(1) 固定長メモリブロックでデータ(メモリセル)を管理する場合、管理データが実データより大きくなりメモリの利用効率が悪くなる。

【0012】(2) 動的に割り付け/開放を行うメモリ域を管理する場合、固定長のデータだけでなく、可変長のデータを必要とすることがある。この場合、固定長および可変長のデータの管理を同じデータ構造を利用して実現できることが好ましく、また、可変長メモリブロックのメモリ管理を行う場合、割り当てと開放の処理時間がそのときと場合によって一定ではなくなる。

【0013】本発明は上記欠点を解消するためになされたもので、その目的はメモリの利用効率の向上及び可変長メモリ管理における処理時間の向上を図ったメモリ管理方式を提供することにある。

【0014】

【課題を解決するための手段】上記目的を達成するために、本発明の請求項1のメモリ管理方式は、マイクロプロセッサからの指令に応じデータのローカルメモリ内のメモリの割り付けと開放を行なうメモリ管理方式において、前記ローカルメモリの一部を固定長の複数のメモリセルに分割し、前記複数のメモリセルにそれぞれビットを対応させるとともに当該メモリセルの使用・未使用の状態を示すビット値として記憶するビットマップを管理データ領域に設け、このビットマップを管理してメモリの割り付けと開放を行なうことを特徴とする。

【0015】本発明の請求項2のメモリ管理方式は、マイクロプロセッサからの指令に応じデータのローカルメモリ内のメモリの割り付けと開放を行なうメモリ管理方式において、前記ローカルメモリの一部を固定長の複数のメモリセルに分割し、前記複数のメモリセルにそれぞれビットを対応させるとともに当該メモリセルの使用・未使用の状態を示すビット値として記憶するビットマップと、前記ビットマップにおける未使用の状態を示すメモリセルの先頭位置およびこの先頭位置のメモリセルに連続する未使用の状態のメモリセルの個数を管理データ領域に設け、メモリの割り付けと開放を行うことを特

徴とする。

【0016】

【作用】本発明(請求項1及び請求項2対応)によれば以下の如き作用を奏する。

(1) このような構成にすることでメモリセルのサイズが小さくなった場合でも、管理用のデータの領域は実データの領域より相対的に大きくなることはない。

【0017】(2) 固定長のメモリ管理のデータ構造を使用して、各メモリセルを連続して割り当てが可能なような上位概念を導入することで、メモリセルのサイズのn倍(ここで、nは1以上でメモリセル全体の個数未満の任意の値)の可変長データを割り当てることが可能となる。

【0018】(3) 可変長のメモリ管理において、固定長メモリ管理のデータに可変長タグ部を追加する。可変長タグ部は、メモリセルの開始とメモリセルの個数で構成される可変長タグの集合体である。可変長タグは、2倍以上のメモリセルを連続して割り当てる場合に、割り当てたメモリセルの開始位置とメモリセルの個数を記録するために使用する。このようなデータ構造によって、可変長メモリ管理におけるメモリ割り当ての処理時間の高速化が図られる。

【0019】

【実施例】以下、本発明の実施例を図を参照して説明する。図1は本発明の第1実施例の構成図であり、本実施例では既に説明した図5に示した従来方式の個々のメモリブロック102にある管理データ114に代えて、実データを統括管理する管理データ領域210を設けた点に特徴がある。

【0020】図1に示すように、固定長のメモリ管理において、(実データの)アドレス部211、(実データの)サイズ部212、(実データ領域の中の各)セルのサイズ(バイト数)部213、ビットマップ部214から構成される管理データは、管理データ領域210に配置し、この管理データを使用して実データの内容を管理する。実データ領域220には、メモリセル221の集合体からなる実データが配置され、管理データとは別の場所に格納されている。

【0021】また、管理データ中のアドレス部211は、実データが存在する場所を示す。サイズ部212は、実データの全体の大きさ(バイト数)を示す。セルのサイズ部213は、実データ領域中の各メモリセル221の大きさ(バイト数)を示す。ビットマップ部214は、実データ領域中の各メモリセル221が使用中であるか、未使用であるかを0か1で表すビットマップの配列を示す。このような構成により管理データ中のビットマップ部214で実データ領域中のメモリセル221の使用状況が把握でき、動的なメモリの管理が可能となる。

【0022】図1に示すように、動的メモリ管理領域の

管理データと実データは、それぞれローカルメモリ 2 上の管理データ領域 210 と実データ領域 220 に配置される。実データは、管理データ領域 210 のアドレス部 211 でその先頭部分が示され、実データ中の各メモリセル 221 は、管理データ領域 210 のセルのサイズ部 213 で示される大きさ（バイト数）である。各メモリセル 221 は、ローカルメモリ 2 上の実データ領域 220 に連続して配置され、その全てを動的なメモリ空間として使用することができる。

【0023】しかして、本実施例の特徴は、図 2 に示すように、実データの各メモリセル 221 を管理データ領域 210 のビットマップ部 214 で利用状態を管理し、さらに、ビットマップ部 214 のビットが 0 のそれに対応する実データのメモリセル 221 が未使用であり、ビットマップ部 214 のビットが 1 の時のそれに対応する実データのメモリセル 221 が使用中であることを表すことである。

【0024】次に、本実施例におけるプロセッサのローカルメモリ上に固定長メモリブロックのメモリ管理をする場合について説明する。今、プログラム 20 が動的メモリの獲得を要求すると、オペレーティングシステム（OS）10 は、管理データ領域 210 中のビットマップ部 214 の各ビットを先頭から順番にチェックし、ビットが 0 である場所を探す。見つければそのビットを 1 にして使用中の状態にし、そのビットに対応する実データのメモリセル 221 をプログラム 20 が利用できるように、アドレスを渡すことによって割り付ける。

【0025】逆に、プログラム 20 が獲得していたメモリを開放する場合、オペレーティングシステム（OS）10 は、プログラム 20 が渡したメモリブロックのアドレスを基に実データ中のメモリセル 221 の位置を計算し、管理データ領域 210 中のこれに対応するビットマップ部 214 のビットを 0 にする事によって未使用状態にし、そのメモリの開放を行う。

【0026】未使用のメモリセル 221 を探すために、ビットマップ部 214 を先頭から 1 ビットずつチェックすると処理に時間がかかるので、バイト単位もしくはプロセッサが 1 回の処理で扱えるビット数を単位としてその単位の全てのビットが 1 でなければその中に未使用のメモリセルがあることが分かる。このようにして処理の高速化を行うことも可能である。

【0027】上述したように、本実施例によれば、動的メモリ管理における管理データ領域 210 は、アドレス部 221、サイズ部 212、セルのサイズ部 213 が一定の大きさを占め、ビットマップ部 214 の大きさが実データ中のメモリセル 221 の個数に比例する。そのため、従来方式における実データの大きさと管理データの大きさの中に占める管理データの割合と比較すると、本実施例におけるその割合は小さく、動的メモリ管理領域 200 を効率的に利用することが可能となる。

【0028】図 3 は本発明の第 2 実施例の構成図であり、本実施例では、管理データ領域 230 と実データ領域 220 によって動的メモリ管理領域 2 が構成され、実データ領域 220 は、上記第 1 実施例と同じように固定長サイズのメモリセル 221 からなる。管理データ領域 230 は、（実データの）アドレス部 231、（実データの）サイズ部 232、（実データの中の各）セルのサイズ（バイト数）部 233、ビットマップ部 234 および可変長タグ部 235 から構成される。

【0029】したがって、本実施例では、可変長のメモリを割り当てる際に、メモリセル 221 のサイズの正数倍の個数の領域を連続して割り当てることができ、このとき何個分のメモリセル 221 を連続して割り当てたかを管理するために、可変長タグ部 235 を使用するものである。

【0030】次に、本実施例におけるプロセッサのローカルメモリ上に可変長メモリブロックのメモリ管理をする場合について説明する。今、プログラム 20 が動的メモリの獲得を要求すると、オペレーティングシステム（OS）10 は、要求に足るだけのサイズ分のメモリセル 221 を獲得できる場所を探す。つまり、管理データ領域 230 中のビットマップ部 234 の各ビットを先頭から順番にチェックし、連続した複数のビットが 0 である所を割り当て可能な場所とする。それが見つければ、その連続したビットを 1 にして使用中の状態にし、そのビットに対応する実データのメモリセル 221 をプログラム 20 が利用できるように、アドレスを渡すことによって割り付ける。さらに、可変長タグ部 235 の可変長タグにメモリセル 221 の開始位置 241 と割り当てたセルの個数 242 を登録する。

【0031】逆に、プログラム 20 が獲得していたメモリを開放する場合、オペレーティングシステム（OS）10 は、プログラム 20 が渡したアドレス情報を基に可変長タグ部 235 の可変長タグをチェックし、そのアドレス情報に一致した可変長タグのセルの開始 241 で示されるメモリセル 221 の開始位置を求め、それに対応するビットマップ部 234 のビットをタグエントリーのセルの個数 242 に登録されている個数分のビットを 0 にすることによって、未使用状態にしてメモリの開放をする。

【0032】上記したように、本実施例では、未使用のメモリセル 221 を探すために、ビットマップ部 234 を 1 ビットずつ順番にチェックする処理としてもよいが、第 1 実施例と同様にバイト単位またはプロセッサが一度に処理できるビット数を単位としてチェックすることで、処理の高速化を行うことができる。

【0033】したがって、本実施例によれば固定長メモリブロック管理をベースとした可変長メモリブロック管理が可能となるので、上記第 1 実施例の効果に加えて、次の効果が得られる。

【0034】すなわち、メモリの割り当ての時のメモリブロックの分割とメモリ開放時のメモリの組み立ての必要がなくなり、処理の高速化を図ることが可能となり、一定の処理時間でメモリの割り当てと開放ができるようになる。

【0035】

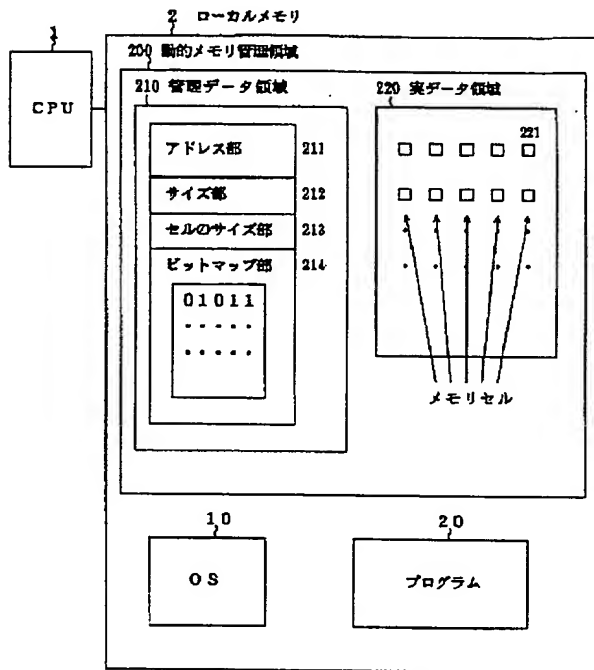
【発明の効果】以上説明したように、本発明によれば、固定長メモリブロックのメモリ管理を行う場合、メモリブロックのサイズやその個数に拘らず効率的なメモリ管理が可能となる。また可変長メモリブロックのメモリ管理を行う場合、処理が一定の速度で高速に実施することが可能となる。

【図面の簡単な説明】

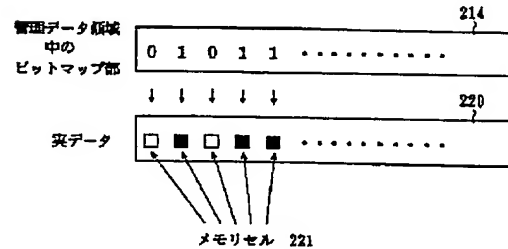
【図1】本発明の第1実施例のメモリ管理方式の構成図。

【図2】図1の管理データ領域中のビットマップ部と実データ中のメモリセルの対応を示す図。

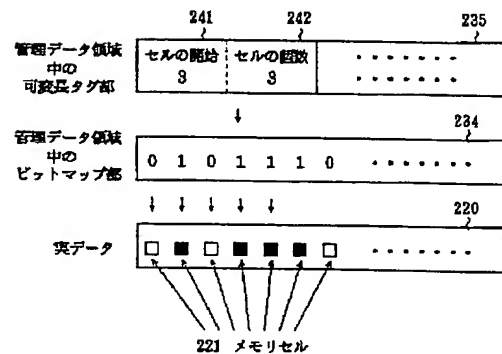
【図1】



【図2】



【図4】



【図3】本発明の第2実施例のメモリ管理方式の構成図。

【図4】図3の管理データ領域中のビットマップ部、可変長タグ部と実データ中のメモリセルの対応を示す図。

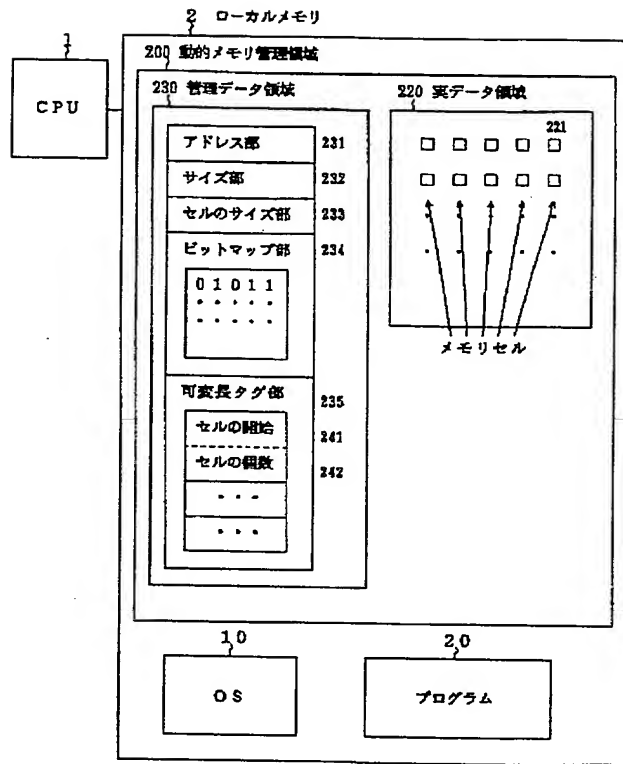
【図5】従来技術による固定長メモリブロックのメモリ管理方式の構成図。

【図6】従来技術による可変長メモリブロックのメモリ管理方式の構成図。

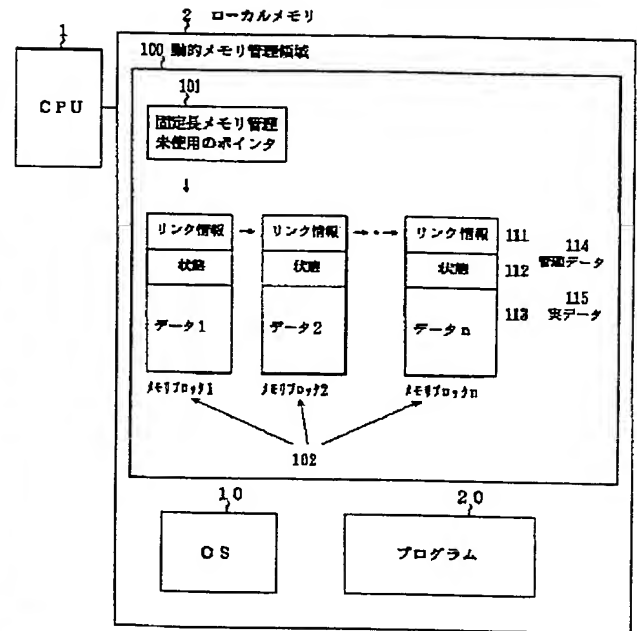
【符号の説明】

- 10 1…マイクロプロセッサ（CPU）、2…ローカルメモリ、10…オペレーティングシステム（OS）、20…プログラム、200…動的メモリ管理領域、210、230…管理データ領域、220…実データ領域、211、231…アドレス部、212、232…サイズ部、213、233…セルのサイズ部、214、234…ビットマップ部、221…メモリセル、235…可変長タグ部、241…セルの開始、242…セルの個数。

【図 3】



【図 5】



【図 6】

